ANSWERS AND QUESTIONS: PROCESSING MESSAGES AND QUERIES*

Catherine N. Ball, Deborah Dahl, Lewis M. Norton, Lynette Hirschman, Carl Weir, Marcia Linebarger

Unisys Corporation
Paoli Research Center
P.O. Box 517, Paoli PA 19301

ABSTRACT

This paper describes issues in adapting the PUNDIT system, designed originally for message processing, to a query-answering system for the VOYAGER application. The resulting system, whose architecture and capabilities are described here, represents a first step towards our goal of demonstrating spoken language understanding in an interactive problem-solving context.

INTRODUCTION

The application which we describe in this paper is a natural language interface to VOYAGER, an expert system which provides navigational assistance for the area between MIT and Harvard[1]. The application is well-suited to exploring the role of natural language understanding in an interactive problem-solving context: the expert system knows enough about its domain to allow extended problem-solving dialogue, e.g. for a user to find a hotel near MIT, find a nearby Italian restaurant, and get directions from the hotel to the restaurant (see below).

Our ultimate goal is to process spoken input, and indeed, dialogue is, to use the terms of [Bib88], a stereotypically *oral* genre. A reasonable strategy for incrementally arriving at this goal is to begin with typed input. This approach allows us to explore general issues in handling dialogue, and further allows us to look at ways in which characteristics of the genre, aspects of the task, the informational structure of the utterance and the discourse context can all be used to limit the search space for spoken language recognition.

The following dialogue illustrates a typical problem-solving session with VOYAGER (note: utterances labelled 'VFE' are contributed by the VOYAGER front-end, discussed shortly).

```
User: Is there a hotel near MIT?
VFE : I'll ask Voyager...
Vgr :   I can't determine if objects are near something.
        I'll try to find the nearest one.
        The nearest hotel to MIT is Marriott.
        Marriott is located at the intersection of Broadway Street and Third
        Street.

User: Is there an Italian restaurant near the hotel?
VFE : I'll ask Voyager...
Vgr :   I can't determine if objects are near something.
        I'll try to find the nearest one.
        The nearest Italian RESTAURANT to Marriott is Bertucci's.
```

1. REPORT DATE **1989**	2. REPORT TYPE	3. DATES COVERED **00-00-1989 to 00-00-1989**

4. TITLE AND SUBTITLE **Answers and Questions: Processing Messages and Queries**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Unisys Corporation,Center for Advanced Information Technology,PO Box 517,Paoli,PA,19301**	8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release; distribution unlimited**

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES **7**	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**			

```
Bertucci's is located at 799 Main Street.
```

```
User: How do I get to the restaurant from the Marriott?
VFE : I'll ask Voyager...
Vgr:   I'll try to find the best way to get from Marriott to Bertucci's.
       Start from Marriott on Broadway Street.
       After you cross Clark Street, take the next left onto Windsor Street.
       There is a traffic light there.
       After you pass School Street on the right, take the next right onto
       Main Street. There is a traffic light there.
       Bertucci's is about forty yards down on your right side.
```

```
User: How long would that take?
(etc.)
```

MESSAGE PROCESSING AND QUESTION-ANSWERING

The PUNDIT natural language processing system was initially developed in the context of message-processing applications. Because PUNDIT is a modular system, typical porting tasks include the creation of a domain-specific lexicon, knowledge base, and semantics rules. Another common feature is basic extensions to handle constructions which are part of the standard written language, but which have not previously appeared in the data (e.g. comparatives, superlatives, address expressions). More interesting are the extensions occasioned by basic differences between messages and face-to-(inter)face conversational interaction.

Basic Differences Between Messages and Voyager Dialogue

To fully characterize these differences requires representative data from naturally-occurring messages and task-oriented dialogue. We have the former; it is our plan to collect the latter, possibly using PUNDIT + VOYAGER as a data collection vehicle. For present purposes, we may highlight some of the observed differences and similarities. Our message corpora consist largely of short narratives in what has been called *telegraphic* style (cf. [GK86]). As a consequence, we find frequent agentless passives, noun-noun compounds, nominalizations, run-on sentences, and zeroing of determiners, subjects, copula, and prepositions. Explicit pronouns and dummy subjects are rare, as are interrogatives, indirect questions, imperatives, and conditionals. Where temporal relations are explicitly marked, they are commonly marked by preposed time adverbials. Modals are rare.

In contrast, our initial corpus of user inputs to VOYAGER shows, not surprisingly, that interrogatives and imperatives are more frequent than statements. The types of zeroing seen in telegraphic narrative do not occur in the corpus[2], nor do nominalizations and run-ons. Passives are rare. Dummy *it* and *there* are common, as are *I* and *you*, and (in the context of questions about restaurants) *they*. Preposing (other than wh-movement) does not occur. Modals are common. Many of these differences are predicted by considering the two genres to be at opposite ends along Biber's textual dimensions of 'involved versus informational production', 'narrative vs. non-narrative concerns', and 'abstract vs. non-abstract information' [Bib88]. We plan to take advantage of such differences to tune the system for the VOYAGER task.

In terms of discourse structure, we might expect there to be radical differences between messages and dialogue, but in fact here there are interesting similarities. The basic interactional unit in VOYAGER dialogue can be characterized as the *request-response pair* (e.g. *Where are you? At MIT.*). A similar request-response structure appears in messages containing labelled discourse segments (or header fields) , e.g. Failed Part: system tape. Cause of Failure: tape was wound backwards. Here, the message originator can be viewed as answering the implicit questions *What part failed? What caused the failure?*

---

[2]Nevertheless, elliptical questions and answers are certainly seen in task-oriented dialogue, e.g. North: [It meant to] ... basically clear up the record. Nields: Did you? North: Tried to. Nields: Then? North: No ... [Scr87]

[Bal89] discusses our approach to handling such structured messages as a series of question/answer pairs, and we were able to extend this approach to dialogue.

The interactional structure of monologue and dialogue is, of course, quite different. To provide the control structure for dialogue, we extended a query front-end (QFE) which had been developed for database query applications. The result is a general dialogue manager which can be used for a variety of interactive applications.

SYSTEM ARCHITECTURE

The system contains four major components: the VOYAGER front-end (VFE), PUNDIT, a query translation and interface module (QTIP), and the VOYAGER expert system. The first three components are currently running on Sun workstations under Quintus Prolog, and VOYAGER, which is written in Lisp, runs on a Symbolics machine. A simplified system flow diagram is shown in Figure 1.

```
      Query/Response
      ----------------------
      |                    |
      |                    |
      |                    |
      |          ___V___   user input
      |          | VFE |   + Context     _____
      |          |     |   ---------->  |        |
      |          |     |   NewContext   | PUNDIT |
      |          |     --- <---------   |_____|          ---
      |          |    | |                                  |   |
      |          |    | | New Info                         | V |
      |          |    | | + Context      _____            | O |
USER  |          |    ------------->  |        | function calls | Y |
      |          |       Response/     | QTIP |  ------...---> | A |
      |          |       errors        |      | response       | G |
      |          |    ---- <--------   |_____| <-----...---   | E |
      |          |    | |                                      | R |
      |          |    | | Vgr resp.      _____              |   |
      |          |    | | + Context     |        |             |___|
      |          |    --------------->  | PUNDIT |
      |          |       NewContext     |        |
      |          |    | <----------    |_____|
      |          |_____|
      |             V
      |  Response/Query   |
      <-------------------
```
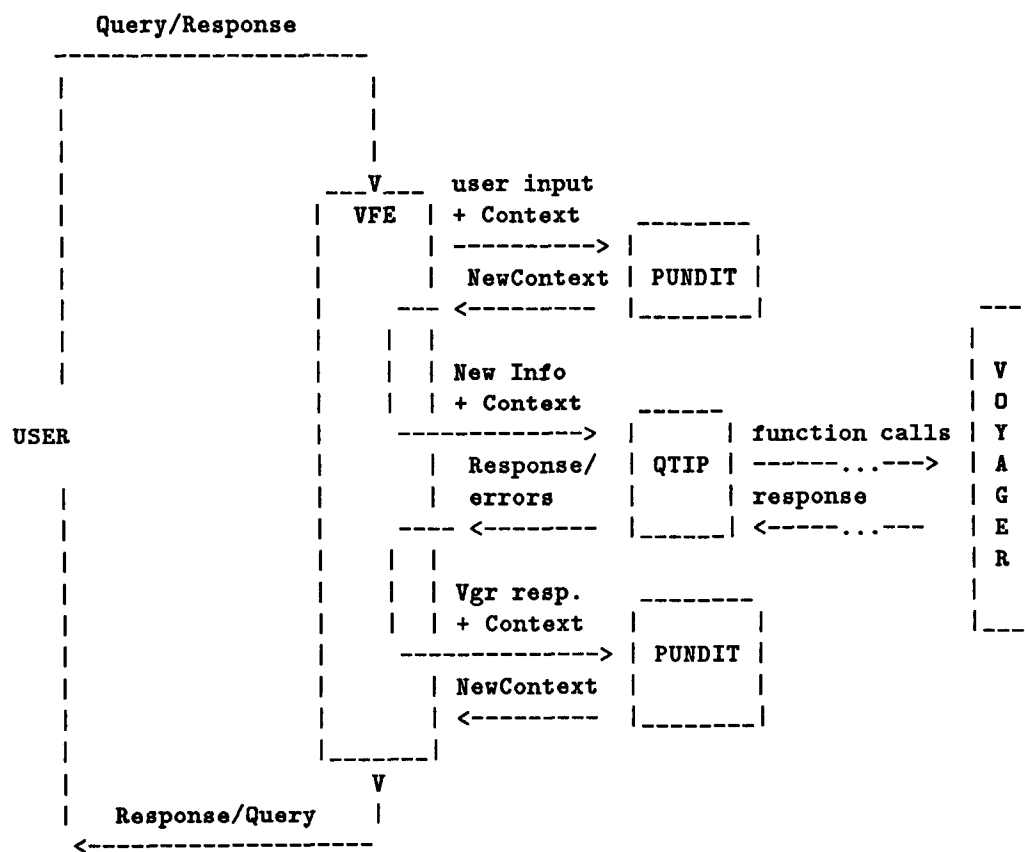
Figure 1: PUNDIT + VOYAGER System Flow

VFE is a dialogue manager, which uses PUNDIT and QTIP as resources to interpret and respond to the user's requests. As discussed below, VOYAGER is also a conversational participant, whose utterances must

62

be analyzed and integrated into the discourse context. VFE administers the turn-taking structure, and maintains a higher-level model of the discourse than that available to PUNDIT. This level of knowledge enables it, for example, to call the parser in different modes, depending on preceding discourse (see below). VFE also keeps track of the current speaker and hearer, so that PUNDIT's Reference Resolution component can correctly interpret *I* and *you*.

PUNDIT, as described in ([HPD+89], [PDP+86], [Dah86]), provides syntactic, semantic, and pragmatic interpretation. The input to PUNDIT is currently text, and the output is a set of semantic representations and other predications representing the discourse context (the DISCOURSE LIST), and a list of entities in focus, ordered by saliency (the FOCUS LIST).

QTIP's function is to translate PUNDIT representations into LISP function calls, to pass these to VOYAGER, and to return VOYAGER's response to VFE. QTIP also incorporates some knowledge about VOYAGER's capabilities which enables it to trap certain types of queries for appropriate action by VFE. For example, VOYAGER cannot answer direction requests with an unspecified starting point, unless it knows where the user is. In this case, QTIP informs VFE that it must elicit the user's location. As another example, VOYAGER cannot answer questions about whether a class of objects is located on a street. QTIP traps such questions, and VFE informs the user:

```
User: Is there a subway station on Church Street?
VFE : Sorry, Voyager can't determine whether something is on a street.
```

QTIP also monitors the state of the machine-machine interface to VOYAGER, and notifies VFE when the link is down or VOYAGER is not loaded; VFE then notifies the user.

The final component is the VOYAGER expert system, a version of which has been made available to us by MIT. VOYAGER includes a generation component, and one of our initial issues was how to deal with its output. Clearly, it would be a poor sort of interactive system that did not allow for ordinary anaphoric and definite reference to entities introduced in the course of the conversation, e.g. to *Royal East* in the example below:

```
A: Is there a restaurant near here?
B: ... The nearest restaurant to MIT is Royal East ...
A: What kind of food does it serve?
```

However, in order to integrate VOYAGER's conversational contribution into the discourse context, we were faced with the choice of (a) modifying VOYAGER to return some semantic representation of its utterances (together with information relevant to focusing), or (b) simply treating VOYAGER as an ordinary conversational participant, and using PUNDIT to analyze what VOYAGER said. The latter proved the most expedient course of action. Though not the most efficient approach in the long run, it has the unexpected benefit of allowing us to look at issues on both sides of the dialogue.

MAINTAINING THE DISCOURSE CONTEXT

Interactive systems must provide for the strategies by which speakers refer to what has been previously discussed, e.g. *one-anaphora*:

Q1. Is there a restaurant that serves seafood near MIT? ... Q2. Is there one near Harvard?

A system which does not handle anaphoric and definite references requires unnatural behaviour on the part of the user, who will be forced to refer to all entities as though they were brand-new in the discourse, e.g. Q2':

Q2'. Is there a restaurant that serves seafood near Harvard?

But it is clearly not sufficient to allow reference only to what the user has said, for the system's response forms part of the user's discourse context, as illustrated above and below.

These points, of course, are not specific to the application or domain, but adhere to language and dialogue. For this reason, the discourse processing facilities of PUNDIT are largely embodied in the domain-independent Reference Resolution module. This module, which required no significant extensions for the VOYAGER application, is capable of handling a variety of referential strategies. In the following annotated dialogue, we illustrate some of the capabilities of Reference Resolution and the importance of understanding both sides of the dialogue.

```
1 User: Is there a restaurant near Harvard ?
2 VFE : I'll ask Voyager...
3 Vgr : I can't determine if objects are near something.
        I'll try to find the nearest one.
        The nearest restaurant to Harvard University is Border Cafe.
        Border Cafe is located at 32 Church Street.
```

Comments: VOYAGER's response is processed by PUNDIT and integrated into the discourse context. As a result, the Border Cafe enters the list of entities in focus, and is thus available for anaphoric reference in (4) below.

```
4 User: What kind of food does it serve ?
5 VFE : I'll ask Voyager...
6 Vgr : Border Cafe serves Mexican food.
```

Comments: it - the user is referring to the Border Cafe, an entity introduced by VOYAGER in turn (3). Reference Resolution quickly finds this entity, which is at the head of the Focus List.

```
7  User: How do I get there?
8  VFE : I'll ask Voyager...
         Where are you?
9  User: At MIT.
10 VFE : Ok. I will tell Voyager...
11 Vgr : MIT is located in the area of 77 Massachusetts Avenue.
         I'll try to find the best way to get from MIT to Border Cafe.
         If your car is on the same side of the street as MIT, start driving.
         Take the next left onto Massachusetts Avenue. There is a traffic light
         there.
         After you cross Garden Street, take the next right onto Church Street.
         Border Cafe is about one eighth mile down on your left side.
```

Comments: there in (7) is interpreted as an anaphoric reference to a salient location. Note the clarification dialogue (VOYAGER will need to know the starting point), where VFE becomes the querier. Since there is a salient wh-query in the context (Where are you?), VFE anticipates a short response[3] and calls the parser and semantics in a special mode. The user's short response is analyzed and bound to the variable in the query (where), and VFE creates a new proposition: The user is at MIT. This information, together with the translation of the original query (How do I get there?) is passed to VOYAGER.

---

[3]For purposes of the present discussion, a short response is one which is not a complete sentence, e.g. What is your name? Mary Smith vs. My name is Mary Smith. See [Hiz79] for a discussion of short responses.

```
12 User: What's the phone number?
13 VFE : I'll ask Voyager...
14 Vgr : The telephone number of Border Cafe is 864-6100.
```

> **Comments:** *the phone number* is a definite reference to an *inferrable* entity (cf. [Pri81]). There is no previously mentioned telephone number, and yet it is a stereotypic assumption that certain classes of objects, e.g. commercial establishments, have phone numbers. This information is encoded in our knowledge base. Reference Resolution looks for previously-mentioned entities that have the property of having phone numbers, and finds the Border Cafe.

```
15 User: Is there a subway stop near the restaurant?
16 VFE : I'll ask Voyager...
17 Vgr : I can't determine if objects are near something.
         I'll try to find the nearest one.
         The nearest subway stop to Border Cafe is Harvard Station.
         Harvard Station is located at the intersection of Massachusetts
         Avenue and Church Street.
```

> **Comments:** *the restaurant* is a definite reference to the Border Cafe. Note that it would not be correct to look for the last explicit mention of a restaurant, for this algorithm would find the restaurant introduced in turn 1: *Is there a restaurant near Harvard?* Instead, Reference Resolution looks for the salient entity of type restaurant, and finds the Border Cafe.

CURRENT COVERAGE AND ISSUES

Current coverage on our training corpus of 174 VOYAGER inputs is 57%; that is, 100 of the utterances are correctly processed by VFE, PUNDIT and QTIP, and are sent to VOYAGER. We plan to increase coverage to 90%, and will then test the system on a more extensive corpus of spontaneous queries, collected by MIT. In addition, we plan to collect and analyze samples of task-oriented dialogue, in order to evaluate the system performance in providing a co-operative interface for interactive problem-solving.

One of the issues which we anticipate addressing is the interpretation of additional types of questions, e.g. short questions. For example, the notions of focus and open proposition (cf. [Pri86]) seem crucial to the interpretation of sequences such as *Are there any bars around MIT? ... How about clubs? ... Restaurants?*.

We also intend to further explore the relationship between the information structure of the question and the answer, in order to optimize the processing of the answer. For example, when VFE asks a clarification question such as *Where are you?*, we construct an open proposition whose variable is typed as a location[4]. We can therefore anticipate a cohesive response from the user in which the open proposition is satisfied by a location. A short response is maximally cohesive (*MIT* or *At MIT*), and we currently anticipate the possibility of this type of response, and call the parser first for an NP fragment, and then (if that fails) for a PP fragment. We could go farther by expecting certain types of prepositional phrases, namely location expressions (e.g. *By subway* would be an incoherent response). For the long response (*I am at MIT*), we may expect the focus to be accented, and the material corresponding to the open proposition to be de-accented. Such expectations should prove useful for speech recognition.

These are not only important research issues, but we believe them to have a direct bearing on the ultimate performance of the system, in terms of user satisfaction and time to accurately complete the task.

---

[4]Intuitively, an open proposition is an expression containing an unbound variable.

References

[Bal89]   Catherine N. Ball. Analyzing explicitly-structured discourse in a limited domain: Trouble and failure reports. In *Proceedings of the DARPA Speech and Natural Language Workshop*, Philadelphia, PA, February 1989.

[Bib88]   Douglas Biber. *Variation Across Speech and Writing.* Cambridge University Press, Cambridge, 1988.

[Dah86]   Deborah A. Dahl. Focusing and reference resolution in PUNDIT. In *Proceedings of the 5th National Conference on Artificial Intelligence*, Philadelphia, PA, August 1986.

[GK86]   Ralph Grishman and Richard Kitteridge, editors. *Analyzing Language in Restricted Domains: Sublanguage Description and Processing.* Lawrence Erlbaum, New Jersey, 1986.

[Hiz79]   Henry Hiz. Difficult questions. In Henry Hiz, editor, *Questions*, pages 211–226. Reidel, Dordrecht, 1979.

[HPD+89] Lynette Hirschman, Martha Palmer, John Dowding, Deborah Dahl, Marcia Linebarger, Rebecca Passonneau, François-Michel Lang, Catherine Ball, and Carl Weir. The pundit natural-language processing system. In *AI Systems in Government Conference.* Computer Society of the IEEE, March 1989.

[PDP+86] Martha S. Palmer, Deborah A. Dahl, Rebecca J. [Schiffman] Passonneau, Lynette Hirschman, Marcia Linebarger, and John Dowding. Recovering implicit information. In *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics*, Columbia University, New York, August 1986.

[Pri81]   Ellen F. Prince. Toward a taxonomy of given-new information. In Peter Cole, editor, *Radical Pragmatics.* Academic Press, New York, 1981.

[Pri86]   Ellen F. Prince. On the syntactic marking of presupposed open propositions. In *Papers from the Parasession on Pragmatics and Grammatical Theory at the 22nd Regional Meeting of the Chicago Linguistic Society*, 1986.

[Ser87]   Federal News Service. *Taking the Stand: The Testimony of Lieutenant Oliver L. North.* Pocket Books, New York, 1987.